2124
41

Kevin W Jameson
148 Edgebank Place NW
Calgary AB
Canada T3A 4L4

August 25, 2004

Trent J Roche
USPTO
Patent Examination Office

In Reply To:	Your office action of June 3, 2004.
Application No:	09/885,077
Application Name:	COLLECTION MAKEFILE GENERATOR
Number of Pages:	22 pages in this response + revised claims pages (73 -84)

Dear Mr.Roche:

This is my response to your office action. I have done my best to show why my present invention is not obvious after the prior art that you cited in the office action.

1 Structure of This Response

2 Introduction

1. To overcome the indefiniteness objections raised by the office action, I have revised my independent claims to more clearly and distinctly claim the subject matter that is my invention.

2. With respect, this response argues against the office claims of obviousness in view of the prior art cited in the office action.

   The prior art cited by the office action is so distant from the present invention that it is irrelevant. Accordingly, I did not cite it myself in the original patent application. This was done on the advice of a USPTO examiner who requested that I avoid sending them "useless or irrelevant" prior art references that they had to review unnecessarily.

   Perhaps my course of action was not the wisest one, because here I must show non-obviousness over the same prior art anyway. As a first time pro se applicant, it is difficult for me to determine whether or not I should include some irrelevant prior art (as perceived by me), if only to show that I did search for relevant prior art.

3 Summary of Revised Claims

I have revised independent claims 1, 10, and 19 to overcome indefiniteness objections, by more clearly citing inventive steps and structures in the claims. In accordance with USPTO patent rules, text in the clauses added to the claims has been underlined. Each of the three claims has been modified as follows:

... comprising the following steps:

(a) receiving a request to generate a makefile for a collection,

(b) accessing collection information for said collection, ~~and~~

(c) classifying said collection information using a collection content classification means, and

(d) generating a makefile for said collection using a collection makefile generator means,

4 My Claims Recite Special Lexicographic Terminology

This section shows that I act as my own lexicographer, and give special meaning to the keyword "collection" and derivative terms such as "collection specifier," "collection content," "collection type," "collection type definition," and "Collection Content Classifier."

4.1 I act as my own lexicographer and define special meanings for key words.

As permitted by patent laws, I act as my own lexicographer and define special meanings for key words in the present application. My non-dictionary meanings of words such as "collection" are commonly misconstrued by patent examiners.

For example, Canadian patent examiners would often perform a simple text search of the prior art using the keyword "collection" to identify possible relevant prior art. However, the prior art found in this way always used the keyword "collection" for its normal dictionary meaning. As you can see, irrelevant prior art patent might use the word "collection" in the dictionary way. Yet the Canadian examiners cited the found search results as relevant prior art anyway. This practice does not seem fair or proper to me, citing irrelevant prior art on the basis of a keyword search.

I respectfully request that USPTO examiners consider my special lexicographic definitions when they cite prior art against the present application. As one USPTO examiner told me, "responding to irrelevant prior art is a waste of everyone's time."

4.2 Definition of "collection"

From the application (page 26), "Collection is a term that refers to the union of a collection specifier and a set of collection content."

In essence, a collection is a software "container" (a software data structure abstraction) that enables automated computer programs to "see, grasp, and manipulate" sets of related computer files.

Technically speaking, collections are inventive data structures whose existence is marked by a special file (collection specifier) that must associate itself with a specific user-defined set of rules (collection data type) for processing the collection. The processing rules are implemented in a special file (collection type definition) that is stored external to the collection, and that can be shared among all collections that associate themselves with that particular data type.

4.3 Definition of "collection specifier"

From the application (page 25), "Collection specifiers contain information about a collection instance."

For example, collection specifiers may define such things as the collection type, a text summary description of the collection, collection content members, derivable output products, collection processing information such as process parallelism limits, special collection processing steps, and program option overrides for programs that manipulate collections.

Collection specifiers are typically implemented as simple key-value pairs in text files or database tables. FIG 3 shows an example physical representation of a collection specifier 102, implemented as a simple text file such as would be used on a typical personal computer filesystem.

4.4 Definition of "collection content"

From the application (page 26), "Collection content is the set of all files and directories that are members of the collection." By convention, all files and directories recursively located within an

identified set of subtrees are usually considered to be collection members. In addition, collection specifiers can contain collection content directives that add further files to the collection membership. Collection content is also called collection membership.

4.5 Definition of "collection type definition"

From the application (page 26), "Collection type definitions are user-defined sets of attributes that can be shared among multiple collections." In practice, collection specifiers contain collection type indicators that reference detailed collection type definitions that are externally stored and shared among all collections of a particular type. Collection type definitions typically define such things as collection types, product types, file types, action types, administrative policy preferences, and other information that is useful to application programs for understanding and processing collections.

4.6 Definition of "collection information"

From the application (page 26), "Collection information is a term that refers to the union of collection specifier information, collection type definition information, and collection content information. "

From the application (page 25), "Collection information is comprised of three major parts: (1) a collection specifier that contains information about a collection instance, (2) a collection type definition that contains information about how to process all collections of a particular type, and (3) optional collection content in the form of arbitrary computer files that belong to a collection."

4.7 Definition of "classification information"

From USPTO application 09/885,076, Collection Content Classifier, filed previously by the applicant, (page 24) "In general, the output classification information answers four questions about a collection. (1) What content does the collection contain? (2) What are the content types of each content file? (3) What processing actions should be carried out on each content file? (4) What processing dependencies exist among content files?"

In other words, classification information is comprised of (1) a list of files that are members of the current collection, (2) a corresponding file type assignment for each file in the membership list, (3) a list of processing actions that should be applied to each file in the list, and (4) a list of processing dependencies among files on the list, that require some files to be processed before other files.

4.8 My inventive collections are not part of the prior art

My inventive collection data structures, and the methods and apparatuses for processing collections, are the subject matter of my patent applications.

Since my applications are based on practical, novel, and non-obvious data structures that are not described in the prior art in any convincing way, my inventions do not read on the prior art, and accordingly, my claims should be allowed.

5 My Claims Recite Specific Inventive Structures

The office action rejects many of my claims and makes the argument that my claims read on the prior art, specifically on FUJII (US Patent 6,301,584) and LU (Application 2002/0147855 A1).

I disagree with the office action because my claims recite many novel inventive data structures and processes that are not shown by the prior art. Neither FUJII nor LU fairly teaches the inventive principles or structures of the present invention. It is not obvious how a person of ordinary skill in the art could reach or infer my inventive principles and structures by reading the cited prior art.

5.1 Summary of Collection Makefile Generator inventive principles and features

- The present Collection Makefile Generator invention discloses a system and method for automatically generating a correct makefile for an arbitrary collection (an inventive data structure) according to policies stored in an external database.

- The main problem solved by the present invention is—*without interactive human participation—how to automatically calculate and generate a precision makefile for managing the efficient application of complex computer command sequences to various collections (an inventive data structure).*

- The main inventive principle of the present invention is *using an external knowledge base of predefined collection type definition information, symbolic process action information, and makefile code templates to dynamically determine and generate a precision makefile for an arbitrary collection (an inventive data structure).*

- The main inventive features of the present invention include *inventive collection data structures, collection specifiers, collection types, collection type definitions, collection content types, collection content action types, and collection content dependency relationships.*

- The main result of the present invention is *an inventive Collection Makefile Generator system and method for automatically and dynamically determining and classifying the content of collections within computer filesystems, for the purpose of automatically generating precision makefiles for processing collections, with no human labor involved.*

5.2 My claims recite specific inventive data structures in Wherein Clauses

My claims are limited by recitation of specific inventive data structures as follows:

- My claims all contain "wherein" clauses that recite my inventive data structures.

- For example, the first of my claims cites "wherein **collections** are data structures comprised of a **collection specifier** and **collection content** containing zero or more collection content files." Each of these special lexicographic terms is specially defined in the application, and is part of one or more inventive data structures that form part of my invention.

- Each wherein clause severely limits one of my claims to the specific inventive collection data structures described in my application.

The office action does not show a convincing line of reasoning that suggests how a person of ordinary skill in the programming arts would be able to reach my inventive data structures and features by reading the cited prior art.

6 Responses to Specific Claim Rejections

The following sections respond to specific office action assertions and rejections.

6.1 Response to indefiniteness, claims 1-27

- Page 2 of the office action states "but it is not clear as to what step or process in the claim provides this solution" as opposed to alternate prior art which discloses the generation of makefiles.

  To overcome this objection, the applicant has revised the independent claims to add a clause that adds more inventive structure to the claims. Furthermore, the applicant submits that the use of the special lexicographic term "collection information" already in the main body of the claim adds further inventive structure to the claim.

- Page 2 of the office action states "no indication is made that any of the steps provide an *automated and scalable* way to generate collection makefiles."

  The applicant submits that automated and scalable methods are strongly associated with the absence of mandatory human labor in processes that are carried out by computers. For example, FUJII requires interactive human labor to define program interfaces and relationships among programs in distributed applications. LU requires the use of human labor to manually create makefiles, which are read and modified by his invention. In contrast, my Collection Makefile Generator requires no human labor whatsoever to dynamically determine and generation a precision makefile that solves NINE major problems in the field of makefile generation (none of which are either addressed or solved by FUJII or LU.)

- Page 3 of the office action states "it is unclear as to what performs the claimed functionality" (in the clauses of the claim, I presume).

  The applicant respectfully submits that the revised claims overcome this objection by more clearly citing more inventive means that accomplish the claimed functionality.

6.2 Response to obviousness, claim 1

- Page 3 of the office action states that FUJII teaches "*accessing collection information for said collection*" by equating my claim words with the phrasing "the makefile generation unit fetches the skeleton source and header file names and the stub source and header filenames from the document management function."

  The terms "collection" and "collection information" refer to inventive data structures that are used by the present invention, but that are completely absent from FUJII in both inherently implied and explicit form.

Therefore FUJII does not fairly teach accessing collection information for a collection, and does not show the inventive data structures used by the present invention.

- Page 4 of the office action states that FUJII teaches "*generating a makefile for said collection*" by equating my claim words with "the makefile generation unit generates a compile command for each source file name acquired."

  I have revised the claims that the office action objects to so that they cite both the use of a collection content classifier means, and the use of a collection makefile generator means. These two additional claim phrases clearly recite inventive means that are disclosed in the present invention, and are clearly absent from FUJII in both inherently implied and explicit form.

  Further, FUJII generates a makefile that merely compiles and links the stub and skeleton files that FUJII generates. This trivial function is a routine technical action that anyone of ordinary skill in the art could do, given the information that FUJII has available at the time of makefile generation—FUJII generated both the source files and filenames in question, and has foreknowledge of the actions (compile and link) and tools (particular languages, compilers, linkers), and operating system beforehand. In contrast, my Collection Makefile Generator has no foreknowledge of any files, filenames, actions, tools, operating systems, or variations in all those things, and yet can still generate a precision makefile for a collection (an inventive data structure). The makefile generation unit in FUJII clearly is not similar to my inventive makefile generator in structure, function, internal methods, or in generation capability.

  Therefore FUJII does not fairly teach generating a makefile for a collection, and does not show the inventive data structures used by the present invention.

- Page 4 of the office action states that FUJII teaches "*wherein collections are data structures comprised of a collection specifier*" by equating my words with "an object ID".

  The office action failed to cite my entire wherein clause, which reads "*wherein collections are data structures comprised of a collection specifier and collection content containing zero or more collection content files, and wherein a collection specifier contains information about a collection instance.*" It is obvious from reading my complete wherein clause that collections are inventive data structures that are comprised of multiple parts (collection specifiers and collection content). Each of the terms collection, collection specifier, and collection content is a special lexicographic term that relates to inventive data structures. Further, it is obvious from the present disclosure that collection specifiers are in no way similar to "an object ID."

  In particular, the office action does not present a credible argument for equating a collection specifier data structure with a simple "object ID" from FUJII. For example, see FIG 3 in the present application, which clearly shows that a collection specifier is a file of information, not a simple integer-valued object ID.

  Therefore FUJII does not fairly teach that collections are inventive data structures that are comprised of a plurality of parts, and does not show the inventive data structures used in the present invention.

- Page 4 of the office action states that FUJII teaches "*collection content containing zero or more collection content files*" by equating my claim words with "the skeleton source and header file names and the stub source and header file names from the document management function."

  The terms "collection content" is a special lexicographic term that relates to inventive data structures used by the present invention. FUJII makes no mention of such inventive data structures, implied or otherwise, and so does not fairly teach the inventive structures of the present invention.

  The present invention provides for dynamically *discovering* collection content files that are members of a collection according to complex and customizable search rules, with no human labor involved whatsoever. In contrast, FUJII teaches the simple retrieval of stub and skeleton source file names from a database where his invention created and stored them, based on input provided by a human operator using the editing unit. Therefore FUJII teaches routine and simple database store and retrieval operations in this regard, nothing more.

  Therefore FUJII does not fairly teach collection content containing zero or more collection content files, and does not show the inventive data structures used by the present invention.

- Page 4 of the office action states that FUJII teaches "*wherein a collection specifier contains information about a collection instance, and wherein collection membership information describes collection content*" by equating my words with "The makefile generation unit acquires interface definition information associated with the application symbol information from the interface information management unit...the skeleton source and header file names and the stub source and header file names from the document management function in accordance with the object Ids."

  The terms "collection specifier" and "collection membership information" are special lexicographic terms that relate to inventive data structures used by the present invention. FUJII does not teach any of my inventive data structures, in either implied or explicit form.

  It is obvious that format, information content, and purpose of a collection specifier file, as shown in FIG 3 of my application, are clearly distinct and not obvious from the interface definition information and structure of the interface information management unit (a simple database) disclosed by FUJII. No person of ordinary skill in the art could reach my inventive structures and methods by reading FUJII (and vice versa). The two inventions and data structures are far apart in concept, structure, function, and operation.

  Therefore FUJII does not fairly teach "wherein a collection specifier contains information about a collection instance, and wherein collection membership information describes collection content," and does not show the inventive data structures used by the present invention.

- Page 4 of the office action agrees that FUJII does not claim to have solved ANY of the nine significant problems in the general makefile generation problem (as the present invention does), but then unreasonably claims that the office action argument has shown that "the structural elements described in the claim have been shown to exist in the

system disclosed by FUJII. Finally, the office action reasons that FUJII must inherently have solved the makefile generator problem anyway.

The applicant has provided many reasons above that show why the logic and claims of the office action are not reasonable or justifiable. In view of the arguments above, no convincing line of reasoning has been shown by the office action to support its claim that *"the structural elements described in the claims have been shown to exist in the system described by FUJII."*

Accordingly, the applicant respectfully submits that the primary supporting reason for the obviousness objections raised by the office action is invalid, and requests that all obviousness objections be withdrawn.

6.3 Response to Obviousness, Claims 2-4, 6

The office action objects to claims 2, 3, 4, and 6 for the same reasons put forth (and responded to) above, that "the structural elements described in the claims have been shown to exist in the system described by FUJII."

In view of the responsive arguments shown above, the applicant respectfully submits that FUJII does not fairly teach the inventive structures of the present application, and requests that all obviousness objections for these claims be withdrawn.

6.4 Response to Obviousness, Claims 10-13, 15, 19-22, and 24

The office action objects to these claims for the same reasons put forth (and responded to) above, that "the structural elements described in the claims have been shown to exist in the system described by FUJII."

In view of the responsive arguments shown above, the applicant respectfully submits that FUJII does not fairly teach the inventive structures of the present application, and requests that all obviousness objections for these claims be withdrawn.

6.5 Response to Prior Art of LU 2002/0147855 A1, Claim 5

Page 4 of the office action states that LU teaches "*the use of virtual platform information*" by equating my words with LU "preferably written in a platform independent language such as, for example, Java..."

However, the office action is misconstruing the meaning of both my terminology "virtual platform information" and the idea of "platform independent programming languages." Those of ordinary skill in the art would obviously see that my virtual platform information—shown clearly in FIG 59-61—is not anything like a platform independent language such as Java. Further, those of ordinary skill in the art would know the difference between a program written in a platform independent programming language (e.g. LU's invention), and the information (e.g. my virtual platform information) that is manipulated by the program.

In view of the responsive arguments shown above concerning LU, and for the arguments concerning FUJII, the applicant respectfully submits that neither FUJII nor LU fairly teach the

inventive structures of the present application. Accordingly, the application respectfully requests that all obviousness objections based on FUJII and LU for this claim be withdrawn.

Applicant respectfully argues that non-obvious combinations of prior art are patentable subject matter, and that even if a novel combination of FUJII + LU taught the subject matter of the present invention—(it does not, of course)—the novel combination would be patentable.

6.6 Response to Obviousness, Claim 7

Page 9 of the office action states that a combination of FUJII + LU teaches *"a product build order"* by equating my words with LU "another make utility will be launched as a new make process which will operate to block further interpretation of makefile wrapper until make utility completes its operations. This prevents the make command from operating out-of-order and causing errors."

The office action is misconstruing the meaning of "product build order."

As disclosed in the present application, a product is a set of files produced from a collection, via some process such as a software build process. Where multiple products are produced from the same collection (the Multiple Product Build Order Problem), it is important that individual products be processed in a proper build order to avoid incorrect results. The present invention solves this problem by providing build order control means in the form of product types and product type build order values, as shown in FIG 50 (multiple products in one collection) and FIG 51.

In contrast, LU does not address or disclose the products, multiple products, the Multiple Product Build Order Problem, product types, product type build orders, or inventive methods or structures for solving the problem. Instead, LU only discloses a simple and routine mechanism for performing "recursive make invocations," a method which has been well-known in the literature and prior art for more than 25 years, ever since Make was invented back in 1975 by Feldman. One aspect of the mechanism is that control flow cannot return to the calling makefile until the recursive call has completed its operation—this is what the office action is misconstruing as a product build order.

LU's use of recursive make invocations to process a subset of makefile targets is not in any way related to the inventive structures disclosed in the present application. Accordingly, the applicant respectfully requests withdrawal of objections for this claim.

6.7 Response to Obviousness, Claim 8

Page 10 of the office action states that neither FUJII nor LU explicitly address or disclose a recognition or solution for the File Build Order Problem, which is the problem of processing particular types of files before other types of files when *calculating*—not executing—a makefile for generation. Yet the office action states that an obvious combination of FUJII + LU "must inherently solve the file build order problem" because "the structural elements described in the claim have been shown to exist in the system described by FUJII modified by LU."

As shown previously, the structural elements cited in the claims of the present application do not exist in FUJII, LU, or in any obvious combination of FUJII + LU. It seems unreasonable to credit FUJII + LU with "inherently solving the problem" when neither of them implicitly or explicitly

recognizes the problem in their disclosures, nor provides any reasonable discussion of a solution in their discussion.

In accordance with the arguments presented above, no convincing line of reasoning has been provided by the office action. The applicant respectfully requests withdrawal of all obviousness objections to this claim.

6.8 Response to Obviousness, Claim 9

Page 10 of the office action agrees that neither FUJII nor LU explicitly address or disclose a either recognition of, or a solution for, the Makefile Parallel Processing Problem, which is the problem of *calculating* an optimal number of makefile targets to be executed in parallel.

The office action states that LU teaches *"the use of parallel limit information to calculate and generate parallel makefile targets"* by equating my words with LU "provides the identity of targets for parallelization and the order in which these targets should be built" and that it would have been obvious to one of ordinary skill in the art that a combination of FUJII + LU would enable some instructions to be *executed* in parallel.

The present Collection Makefile Generator invention is concerned with the *calculation and generation* of parallel makefile targets, not with the execution of parallel targets. In fact, neither FUJII nor LU disclose any explicit method or structure for calculating and generating parallel makefile targets for processing a collection (an inventive data structure) or its collection contents (a special lexicographic term).

Since neither FUJII nor LU explicitly discuss the calculation and generation of parallel makefile targets, reaching the inventive structures and methods and results of the present invention by reading FUJII and LU would clearly NOT have been obvious to a person of ordinary skill in the art.

Further, previous arguments above have shown that "the structural elements described in the claim" do NOT exist in the imaginary "system disclosed by FUJII and modified by LU" that the office action supposes. Lastly, the imaginary system disclosed by FUJII modified by LU does not mention, disclose, or inherently provide a solution to the Makefile Parallel Processing Problem as described and solved by the present invention.

6.9 Response to Obviousness, Claims 14, 16-18, 23, and 25-27

The office action objects to these claims for the same reasons put forth (and responded to) above, that "the structural elements described in the claims have been shown to exist in the system described by FUJII modified by LU."

In view of the responsive arguments shown above, neither FUJII, LU, nor a combination of FUJII modified by LU fairly teach the inventive structures, functions, methods, or results of the present application. The applicant respectfully requests that all obviousness objections for these claims be withdrawn.

The office action has not shown any convincing reasons why my inventive data structures, methods, and apparatuses read on the prior art. Therefore, it is unreasonable to expect that a person of ordinary skill in the art would find it obvious to invent collection data structures or the

associated methods and apparatus of the present application, by reading only the prior art cited by the office action.

I have shown how the office action has misconstrued the special lexicographic meanings of "collection" (or other derived "collection" terms that represent my inventive structures), and how FUJII does not fairly teach the inventive principles, methods, or structures of my invention.

Accordingly, the applicant respectfully requests that this rejection be withdrawn.

7 Summary of the Present Invention

The Applicant requests reconsideration and withdrawal of all prior art objections because there are many reasons that the present invention is not obvious after the cited prior art. A supporting analysis is presented in the sections that follow.

7.1 Summary of the present Collection Makefile Generator Invention

The present invention discloses a system and method for *automatically calculating and generating a precision parallel makefile for managing the efficient application of complex computer command sequences to collections.*

The main problem solved by the present invention is *how to automatically calculate and generate a precision parallel makefile for managing the efficient application of complex computer command sequences to collections, with no human labor involved.*

The main inventive principle of the present invention is *the use of inventive data structures (collections, and an external database of generation information) and a smart generator program to support the dynamic calculation and generation of precision makefiles that solve many simultaneous and different subproblems of the general Collection Makefile Generation Problem.*

The main inventive features of the present invention are *collections, collection types, collection type definitions, product types, product type definitions, product build order information, file build order information, virtual platform information, parallel execution groups, makefile fragment templates, makefile services, and a smart makefile generator program.*

The main result of the present invention is *a system and method for automatically calculating and generating precision parallel makefiles for managing the efficient application of complex computer command sequences to collections, and for solving many difficult subproblems of the overall Collection Makefile Generation Problem*—with no human labor involved.

8 Response to Prior Art of FUJII

The Applicant requests reconsideration and withdrawal of all prior art objections because there are many reasons that the present invention is not obvious after the cited prior art. A supporting analysis is presented below.

8.1 Summary of FUJII

FUJII discloses a system and method for *supporting the design of a plurality of programs distributed on a network and operated through the communication between the programs.*

The main problem solved by FUJII is *enabling human operators to more conveniently design and construct distributed applications that use an ORB (object request broker) to locate and invoke network programs that comprise the distributed application.*

The main inventive principle of FUJII is *the use of a system that supports human operators by providing support for the graphical connection and specification of interfaces among multiple programs that comprise a distributed application.*

The main inventive feature of FUJII is *a system and a database of interface specification information that generates stub and skeleton source files automatically from human-provided interface information that is stored in the database.*

The main result of FUJII is *a system and method for supporting the design of a plurality of programs distributed on a network, and which generates simple stub and skeleton source files from a database of human-provided interface specification information for the programs so generated.*

The claims of the present invention do not read on the FUJII disclosure in an obvious way, as the following points demonstrate.

8.2 FUJII solves a different problem.

FUJII solves the problem of *enabling human operators to more conveniently design and construct distributed applications that use an ORB (object request broker) to locate and invoke network programs that comprise the distributed application.*

The present invention solves the problem of *how to automatically calculate and generate a precision parallel makefile for managing the efficient application of complex computer command sequences to collections, with no human labor involved.*

The two problems are very different, making the present invention not obvious from FUJII.

8.3 FUJII does not show the inventive principle of my invention.

FUJII does not show, nor teach toward, the inventive principle of the present invention.

The main inventive principle of FUJII is *the use of a system that supports human operators by providing support for the graphical connection and specification of interfaces among multiple programs that comprise a distributed application.*

The main inventive principle of the present invention is *the use of inventive data structures (collections, and an external database of generation information) and a smart generator program to support the dynamic calculation and generation of precision makefiles that solve many simultaneous and different subproblems of the general Collection Makefile Generation Problem.*

The two inventive principles of operation are very different, making the present Collection Makefile Generator invention not obvious from FUJII.

8.4 FUJII does not show the inventive features of my invention.

FUJII does not show, or teach toward, any of the inventive features recited in the claims of the present Collection Makefile Generator invention.

The main inventive feature of FUJII is *a system and a database of interface specification information that generates stub and skeleton source files automatically from human-provided interface information that is stored in the database.*

The main inventive features of the present invention are *collections, collection types, collection type definitions, product types, product type definitions, product build order information, file build order information, virtual platform information, parallel execution groups, makefile fragment templates, makefile services, and a smart makefile generator program.*

The number and character of these distinct inventive features show clearly that the present Collection Makefile Generator invention is not obvious from FUJII.

8.5 FUJII does not show the unexpected results of my invention.

The main result of FUJII is *a system and method for supporting the design of a plurality of programs distributed on a network, and which generates simple stub and skeleton source files from a database of human-provided interface specification information for the programs so generated.*

The main result of the present invention is *a system and method for automatically calculating and generating precision parallel makefiles for managing the efficient application of complex computer command sequences to collections, and for solving many difficult subproblems of the overall Collection Makefile Generation Problem*—with no human labor involved.

Furthermore, the act of *generating precision makefiles for complex command sequences* enables collection-aware programs to dynamically manipulate collections (through the automatic generation and execution of precision makefiles), with no human labor involved.

This unexpected result—being able to manipulate collections using precision makefiles—frees human operators from having to interactively specify multiple operations to process the same set of collections.

FUJII does not teach this unexpected result. This demonstrates that the unexpected results of the present Collection Makefile Generator invention are not obvious from FUJII.

8.6 FUJII teaches away from my invention regarding need for human users.

FUJII teaches a system and method for *enabling human operators to more conveniently design and construct distributed applications.*

FUJII therefore teaches that human users are required in his system.

In contrast, the present invention teaches about a totally automated Collection Makefile Generator invention for use with totally automated collection processing systems, which require no human labor at all.

FUJII therefore teaches away from the present Collection Makefile Generator invention concerning the need for human users, which makes the present invention not obvious from FUJII.

8.7 FUJII does not suggest modifications to meet the present claims.

FUJII does not teach—or even suggest—that his design system for client/server programs could be modified in an obvious way to meet the claims of the present invention.

Accordingly, it is reasonable to conclude that the present Collection Makefile Generator invention was not contemplated by, nor was obvious to FUJII, and is not obvious to a person of ordinary skill in the art.

8.8 No convincing reasoning for obviousness has been presented.

The points above show that FUJII solves a different problem and does not show any of the inventive principles, features, or results of my invention.

In view of these points, the office action has not presented a convincing line of reasoning for explaining why a person of ordinary skill in the art would find the claimed subject matter of the present invention to be obvious after the cited prior art.

The lack of a clear line of reasoning in the office action suggests that the claim of obviousness cannot be easily justified, and that the present invention is not obvious from the cited prior art.

8.9 The present invention solves an unrecognized problem.

The present Collection Makefile Generator invention solves the previously unrecognized problem of *enabling automated computer programs to automatically calculate and generate precision parallel makefiles for inventive collection data structures.*

Neither the prior art nor the industry literature has previously recognized the various (nine) Collection Makefile Generation problems for inventive *collection data structures*. This is because the prior art is not aware of inventive collection data structures themselves.

In contrast, the present invention is one of a series of novel inventions that are bringing the advantages of collections to the art in the form of patent disclosures and commercial software products.

Since the prior art does not show awareness of either the main problem solved by the present invention, nor of its many inventive features, it is reasonable to conclude that the present invention is not obvious from the prior art.

8.10 The present invention solves a long-felt and unsolved need.

The present invention helps to increase the power, convenience, and automation of computer programs by providing them with a general way to work with inventive *collection data structures* that contain various kinds of computer files.

The industry has long felt the need for a convenient way of representing and manipulating groups of related computer files (which I store in inventive collection data structures). But no general, user-extensible, and user-customizable solution has been known to the prior art. The present Collection Makefile Generator invention helps to solve this long-felt industry need.

It follows that the present invention is not obvious from the prior art, else alternative solutions would already be known and be in common use within the programming arts.

8.11 The present invention shows unappreciated advantages.

The present invention shows several advantages that are not obvious to, and are not appreciated by, those of ordinary skill in the art.

For example, the present invention enables the construction of automated collection processing systems *that require no human labor* to run automated computational processes on large systems of collections.

For another example, a practical application of the present invention is to help solve the ubiquitous multiplatform software build problem.

It follows that the present invention is not obvious from the prior art, because the prior art does not teach these unappreciated advantages.

8.12 The present invention has not been implemented before.

The prior art lacks any mention of the present invention *for inventive collection data structures*, and lacks any implementations of the present invention.

It follows that the present invention is not obvious to the prior art, else those of ordinary skill in the art would have implemented the present invention by now, in order to capture the many benefits of the present invention.

8.13 The present invention is contrary to the prior art.

The prior art clearly teaches application program designs that require human operators—using multiple, tedious operations—to provide various inputs to computer programs, and to tell application programs which files to use.

For example, FUJII, in the prior art cited by the office action, teaches the use of human users to provide input program interface and design criteria.

In contrast, the present invention is intended for use by totally automated computer systems, with no human labor required.

It follows that the present invention is not obvious to those of ordinary skill in the art, because the inventive principles of the present invention, *including the principle of no human labor required,* are contrary to the teachings of the prior art, such as FUJII.

9 Response to Prior Art of LU

The Applicant requests reconsideration and withdrawal of all prior art objections because there are many reasons that the present invention is not obvious after the cited prior art. A supporting analysis is presented below.

9.1 Summary of LU

LU discloses a system and method for *enabling humans to process existing makefile commands in parallel on a multiprocessor computer.*

The main problem solved by LU is *enabling humans to more easily use multiple CPUs on multiprocessor computers when executing their existing makefiles.*

The main inventive principle of LU is *the use of an invasive client-server processing system to move processing of existing makefile commands from one processor to another in a multiprocessor computer system.*

The main inventive features of LU are *a makefile wrapper, a command client that intercepts existing makefile commands, and a command server that buffers and eventually executes buffered (and existing) makefile commands.*

The main result of LU is *a system and method for enabling humans to perform existing makefile commands in parallel on a multiprocessor computer.*

The claims of the present invention do not read on the LU disclosure in an obvious way, as the following points demonstrate.

9.2 LU solves a different problem.

LU solves the problem of *enabling humans to more easily use multiple CPUs on multiprocessor computers when executing their existing makefiles.*

The present invention solves the problem of *how to automatically calculate and generate a precision parallel makefile for managing the efficient application of complex computer command sequences to collections, with no human labor involved.*

The two problems are very different, making the present invention not obvious from LU.

9.3 LU does not show the inventive principle of my invention.

LU does not show, nor teach toward, the inventive principle of the present invention.

The main inventive principle of LU is *the use of an invasive client-server processing system to move processing of existing makefile commands from one processor to another in a multiprocessor computer system.*

The main inventive principle of the present invention is *the use of inventive data structures (collections, and an external database of generation information) and a smart generator program to support the dynamic calculation and generation of precision makefiles that solve many simultaneous and different subproblems of the general Collection Makefile Generation Problem.*

The two inventive principles of operation are very different, making the present Collection Makefile Generator invention not obvious from LU.

9.4 LU does not show the inventive features of my invention.

LU does not show, or teach toward, any of the inventive features recited in the claims of the present Collection Makefile Generator invention.

The main inventive features of LU are *a makefile wrapper, a command client that intercepts existing makefile commands, and a command server that buffers and eventually executes buffered (and existing) makefile commands.*

The main inventive features of the present invention are *collections, collection types, collection type definitions, product types, product type definitions, product build order information, file build order information, virtual platform information, parallel execution groups, makefile fragment templates, makefile services, and a smart makefile generator program.*

The number and character of these distinct inventive features show clearly that the present Collection Makefile Generator invention is not obvious from LU.

9.5 LU does not show the unexpected results of my invention.

The main result of LU is *a system and method for enabling humans to perform existing makefile commands in parallel on a multiprocessor computer.*

The main result of the present invention is *a system and method for automatically calculating and generating precision parallel makefiles for managing the efficient application of complex computer command sequences to collections, and for solving many difficult subproblems of the overall Collection Makefile Generation Problem*—with no human labor involved.

Furthermore, the act of *generating precision makefiles for complex command sequences* enables collection-aware programs to dynamically manipulate collections (through the automatic generation and execution of precision makefiles), with no human labor involved.

This unexpected result—being able to manipulate collections using precision makefiles—frees human operators from having to interactively specify multiple operations to process the same set of collections.

LU does not teach this unexpected result. This demonstrates that the unexpected results of the present Collection Makefile Generator invention are not obvious from LU.

9.6 LU teaches away from my invention regarding need for human users.

LU teaches a system and method for *enabling humans to process existing makefile commands in parallel on a multiprocessor computer.*

LU therefore teaches that human users are required in his system, to provide inputs such as existing makefiles and other information required for his invention to work.

In contrast, the present invention teaches about a totally automated invention for use with totally automated collection processing systems, which require no human labor at all.

LU therefore teaches away from the present Collection Makefile Generator invention concerning the need for human users, which makes the present invention not obvious from LU.

9.7 LU does not suggest modifications to meet the present claims.

LU does not teach—or even suggest—that his object-oriented web search system could be modified in an obvious way to meet the claims of the present invention.

Accordingly, it is reasonable to conclude that the present Collection Makefile Generator invention was not contemplated by, nor was obvious to LU, and is not obvious to a person of ordinary skill in the art.

9.8 No convincing reasoning for obviousness has been presented.

The points above show that LU solves a different problem and does not show any of the inventive principles, features, or results of my invention.

Accordingly, the office action has not presented a convincing line of reasoning for explaining why a person of ordinary skill in the art would find the claimed subject matter of the present invention to be obvious after the cited prior art.

The lack of a clear line of reasoning in the office action suggests that the claim of obviousness cannot be easily justified, and that the present invention is not obvious from the cited prior art.

9.9 The present invention solves an unrecognized problem.

The present Collection Makefile Generator invention solves the previously unrecognized problem of *enabling automated computer programs to automatically calculate and generate precision parallel makefiles for inventive collection data structures.*

Neither the prior art nor the industry literature has previously recognized the various (nine) Collection Makefile Generation problems for inventive *collection data structures*. This is because the prior art is not aware of inventive collection data structures themselves.

In contrast, the present invention is one of a series of novel inventions that are bringing the advantages of collections to the art in the form of patent disclosures and commercial software products.

Since the prior art does not show awareness of either the main problem solved by the present invention, nor of its many inventive features, it is reasonable to conclude that the present invention is not obvious from the prior art.

9.10 The present invention solves a long-felt and unsolved need.

The present invention helps to increase the power, convenience, and automation of computer programs by providing them with a general way to work with inventive *collection data structures* that contain various kinds of computer files.

The industry has long felt the need for a convenient way of representing and manipulating groups of related computer files (which I store in inventive collection data structures). But no general,

user-extensible, and user-customizable solution has been known to the prior art. The present Collection Makefile Generator invention helps to solve this long-felt industry need.

It follows that the present invention is not obvious from the prior art, else alternative solutions would already be known and be in common use within the programming arts.

9.11 The present invention shows unappreciated advantages.

The present invention shows several advantages that are not obvious to, and are not appreciated by, those of ordinary skill in the art.

For example, the present invention enables the construction of automated collection processing systems *that require no human labor* to run automated computational processes on large systems of collections.

For another example, a practical application of the present invention is to help solve the ubiquitous multiplatform software build problem.

It follows that the present invention is not obvious from the prior art, because the prior art does not teach these unappreciated advantages.

9.12 The present invention has not been implemented before.

The prior art lacks any mention of the present invention *for inventive collection data structures*, and lacks any implementations of the present invention.

It follows that the present invention is not obvious to the prior art, else those of ordinary skill in the art would have implemented the present invention by now, in order to capture the many benefits of the present invention.

9.13 The present invention is contrary to the prior art.

The prior art clearly teaches application program designs that require human operators—using multiple, tedious operations—to provide various inputs to computer programs, and to tell application programs which files to use.

For example, LU, in the prior art cited by the office action, teaches the use of human users to provide input *existing* makefiles for use by the invention, a makefile wrapper program, and such values as "serverPortNumber" and "maxNumberOfProcesses."

In contrast, the present invention is intended for use by totally automated computer systems, with no human labor required.

It follows that the present invention is not obvious to those of ordinary skill in the art because the inventive principle of the present invention, *including the principle of no human labor required,* is contrary to the prior art.

10 Request for withdrawal of all USPTO objections.

To overcome all indefiniteness objections, I have modified all independent claims to recite additional inventive structure in the claim phrases.

To overcome all obviousness objections, I have provided reasoning that explains why the present invention is not obvious after the prior art that was cited by the examiner.

Specifically, the applicant respectfully submits that the present application is not obvious after FUJII or LU, and that all claims now comply with USPTO patent rules.

Accordingly, the applicant respectfully requests reconsideration and withdrawal of all objections.

I hope that this response will allow continued prosecution of my patent application. If you require more changes, I would be happy to carry them out.

Respectfully yours,

Kevin W Jameson
Inventor Pro Se